AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer implemented method for fault tolerance, load balance and failover of CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references; ~~invoking a cluster contained in a context;~~

in response to a request from a client that invokes a cluster, performing a load balance by having the name service ~~to~~ select an object server located in the invoked cluster;

appending a cluster component to the invoked cluster to provide failover upon failure of the object server;

forwarding a selected object reference to a client upon completion of the load balance; and

~~communicating~~ permitting the client to communicate with ~~a~~ the server associated with the selected object reference which was forwarded to the client~~.~~, wherein the fault tolerance, the load balance and the failover are performed transparently.

2. The method of claim 1, said invoking step comprising the step of:

binding to the server using an IP Address and port number contained in the specific object reference.

3. (Currently Amended) A computer implemented method for fault tolerance, load balance and failover of CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references;

in response to a request from a client that invokes a cluster,

performing a load balance by having the name service select an object server located in the invoked cluster;

appending a cluster component to the invoked cluster to provide failover upon failure of the object server;

forwarding a selected object reference to a client upon completion of the load balance;

permitting the client to communicate with the server associated with the selected object reference which was forwarded to the client by binding to the server using an IP Address and port number contained in the specific object reference, wherein the fault tolerance, the load balance, and the failover are performed transparently. ~~The method of claim 2, further comprising the steps of:~~

indicating to a user whether bind interceptors are in use;

providing the user with a class having relevant methods if bind interceptors are in use; and

specifying the class such that the class contains the most relevant methods, said specification being performed at a discretion of the user.

4. (Original) The method of claim 3, further comprising the steps of:

checking the bind interceptors if the object server fails; and

selecting an alternative server if a bind interceptor contains a predetermined method; said selection being performed by the user upon entry of the predetermined method by the user.

5. (Original) The method of claim 4, further comprising the steps of:

intercepting a cluster component of the object server which failed based on the bind interceptor;

invoking a load balance algorithm of the cluster via the bind interceptor to select and return a new object reference located in the cluster to the client; establishing communications with the client and a server of the new object

reference; and

marking the failed object server to indicated failure thereof.

6. (Original) The method of claim 5, further comprising the step of:

removing the marked failed object server from the cluster.

7. (Original) The method of claim 4, wherein the predetermined method is Bind_Failed.

8. (Original) The method of claim 3, wherein the most relevant methods are one of Bind,
Bind_Succeeded and Bind_Failed.

9. (Currently Amended) The method of claim 1, further comprising the step of:

specifying a load balance algorithm upon ~~creation of a~~ establishing the naming service cluster to perform name service load balancing of object references contained within the clusters.

10. (Original) The method of claim 1, wherein said load balancing is performed based on a predetermined method.

11. (Original) The method of claim 4, wherein the predetermined method is a Round robin load balancing algorithm.

12. (Original) The method of claim 1, wherein said load balancing is performed based on a predetermined method.

13. (Original) The method of claim 12, wherein the predetermined method is a Round robin load balancing algorithm.

14. (Currently Amended) The method of claim 1, wherein ~~each cluster contains an~~ the object binding table ~~which~~ of each cluster contains object references;

wherein each object server reference represents a single server.

15. (Currently Amended) A computer implemented method for fault tolerance, load balance and failover of CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references;

setting a flag in a file to activate the implicit clustering;

in response to invoking a cluster contained in a context having clusters; performing a load balance by having the name service ~~to~~ select an object server located in the clusters;

forwarding a selected object reference to a client upon completion of the load balance; and

communicating with the server associated with the selected object reference which was forwarded to the client~~.~~, wherein the fault tolerance, the load balance and the failover are performed transparently.

16. (Original) The method of claim 15, wherein the file is a configuration file.

17. (Original) The method of claim 15, said invoking step comprising:

binding to the server using an IP Address and port number contained in the specific object reference.

18. (Original) The method of claim 15, wherein said load balancing is performed based on a predetermined method.

19. (Original) The method of claim 18, wherein the predetermined method is a Smart Round Robin load balancing algorithm.

20. (Original) The method of claim 15, wherein object reference binding having identical names are clustered together in common clusters such that a common group of object reference binders servers is created.

21. (Original) The method of claim 20, further comprising the step of:

specifying a load balance algorithm to perform load balancing of object references contained within the common group of group of object reference binders.

22. (Original) The method of claim 21, wherein initially the load balance algorithm is Smart Round Robin.

23. (New) A computer implemented method for transparently load balancing CORBA object servers, comprising the steps of:

establishing name service clusters for the object servers which each contain a unique object binding table that contains object server references;

in response to a request from a client that invokes a cluster, the name service for the cluster performs a load balance to select an object server located in the invoked cluster which can handle the request;

forwarding a selected object reference to a client upon completion of the load balance; and

permitting the client to communicating with the server associated with the selected object reference that was forwarded to the client.

24. (New) The method of claim 23 further including the step of appending a cluster component to the invoked cluster, such cluster providing information to the client as to a failover server to be accessed upon failure of the object server.